

A DSP Based H.264 Decoder for a Multi-Format IP Set-Top Box

F. Pescador, M. J. Garrido, C. Sanz, E. Juárez, D. Samper.
Universidad Politécnica de Madrid. Spain.
{pescador, matias, cesar, ejuarez, dsamper}@sec.upm.es

Abstract-- In this paper, the implementation of a digital signal processor (DSP) based H.264 decoder for a multi-format set-top box is described. Baseline and Main profiles are supported. Using several software optimization techniques, the decoder has been fitted into a low-cost DSP. The decoder alone has been tested in simulation, achieving real-time performance with a 600 MHz system clock. Finally, it has been integrated in a multi-format IP set-top box using a commercial development board based on the DSP @ 600 MHz. Tests in a real environment have been performed using this board with good results.

I. INTRODUCTION

In the last years, new video coding standards [1] [2] have been adopted allowing more data compression but increasing the complexity for both encoders and decoders as well [3]. With the latest generation of Digital Signal Processors (DSPs) [4]-[6], very flexible decoders can be implemented at a relative low cost. The complexity of an H.264 decoder may increase by a factor up to 2 regarding to MPEG-4 SP [3], which in turn is more complex than MPEG-2. Thus, a real-time H.264 standard definition DSP-based decoder is hard to obtain [7]-[9].

In this paper, the implementation of an H.264 decoder based on a low-cost DSP and its integration on a multi-format IP set-top box (STB) is described. The most important challenge is to integrate all the IP STB tasks in a low-cost DSP. Section II explains the decoder implementation. Section III describes the tests performed and their results. Section IV describes the decoder integration into the STB. Finally, section V is devoted to the conclusions.

II. DECODER IMPLEMENTATION

The decoder has been implemented on a low-cost fixed point video-oriented DSP [6]. Basic Profile (BP) and Main Profile (MP)¹ of H.264 standard [2] at level 3 have been implemented. The starting point has been a standard compliant raw-C decoder fully tested first in a PC environment and then moved to the DSP. This initial code was optimized to increase the execution speed in two orders of magnitude.

In Fig. 1, a simplified flow diagram of the decoding process for a slice unit is shown. After decoding the NAL header, the NAL unit content is identified as a slice or another syntax element. Afterwards, the slice header is processed; a loop is

performed to decode every MB. Virtually all computational load is consumed in the slice decoding.

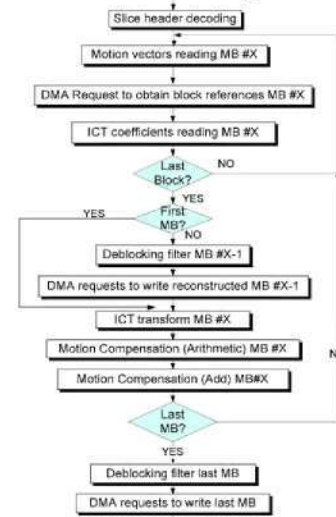


Fig.1. Simplified flow diagram of the decoding process.

Several optimization steps have been developed to improve the decoder performance in speed:

- Frequent arithmetic operations have been coded using intrinsic (pseudo-assembler) instructions. The same has been done with the core of the Deblocking Filter (DF), Integer Cosine Transform (ICT), Context Adaptive Variable Length Coding (CAVLC) and Motion Compensation (MC).
- The Context Adaptive Binary Arithmetic Coding (CABAC) core has been optimized, encoded in assembly language and parallelized by hand.
- The decoding loop has been reorganized to eliminate the DMA waits associated with the request of the reference macroblocks. DF for MB#X-1 is executed after the requests of reference macroblock MB#X. More instructions are executed between DMA data requests and MC.
- Data needed for DF are saved in internal buffers to reduce data movement as it was presented in [10].
- Data movement and decoding operations have been parallelized as shown in Fig. 2.

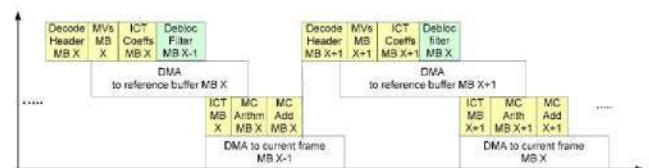


Fig.2. Parallelization of data movement with the decoding operations.

A set of simulation tests has been carried out to verify the decoder and to measure its performance. Actual DVD movies like “Star Wars: episode I” and “Finding Nemo” and a football sequence from a digital TV channel have been used to generate both BP and MP H.264 test streams². The test-bench is shown in Fig. 3. First, a test stream is read from a file on a picture basis and written into a stream buffer allocated in external memory. Then, the decoder reads the stream from this memory decodes it and writes the decoded picture into a file.



Fig.3. Test-bench used to profile the decoder.

Table I contains the profiling results, in average clock cycles per frame, for the decoder and its main parts: CABAC (MP), CAVLC (BP), ICT+MC, DF and others. The CPU% row shows the percentage of CPU load spent by the decoder with a 600 MHz system clock.

TABLE I.
H.264 DECODER PERFORMANCE IN SIMULATION.

| # CYCLES×10 ⁶ | NEMO | | STAR WARS | | FOOTBALL | | NEMO1M | |
|--------------------------|------|------|-----------|------|----------|------|--------|------|
| | BP | MP | BP | MP | BP | MP | BP | MP |
| DECODER | 18.5 | 23.3 | 17.6 | 23.4 | 18.0 | 23.3 | 14.9 | 22.0 |
| CAVLC/CABAC | 5.1 | 8.2 | 4.9 | 8.0 | 5.1 | 7.9 | 3.6 | 7.1 |
| ICT+MC | 6.0 | 7.2 | 6.0 | 7.5 | 5.9 | 7.7 | 5.3 | 6.5 |
| DF | 5.9 | 6.1 | 5.2 | 6.0 | 5.2 | 5.8 | 3.9 | 6.4 |
| OTHERS | 1.5 | 1.8 | 1.5 | 1.9 | 1.8 | 1.9 | 2.1 | 2.0 |
| CPU% | 77.1 | 97.0 | 73.3 | 97.5 | 75.0 | 97.0 | 62.1 | 91.6 |

III. INTEGRATION INTO THE IP STB

The decoder has been integrated into an IP STB [11] and tested using a board [12] based on the DSP @ 600 MHz. The IP STB has a multi-task software architecture that has been built using the RF5 [13] model (see Fig. 4). A multi-task real-time kernel schedules the tasks execution and allows inter-tasks communication. The H.264 decoder has been embedded in the video decoding task. The test-bench can be seen in Fig. 5. A commercial encoder [14] generates the test sequences encapsulated in MPEG-2 Transport Stream over IP³. The board decodes and presents the audio and video information on a TV display. In Table II, the percentage of CPU load spent by the decoder and by the overall system is given. These data have been measured using an internal DSP timer.

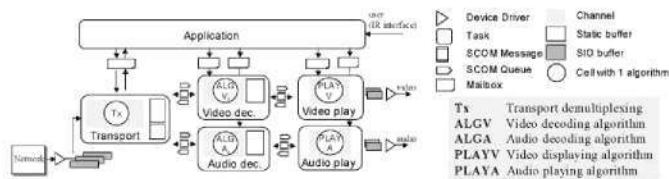


Fig.4. Block diagram of the software architecture (notation details in [13]).

² Length: 100 pictures. Format: 720×576 pels @ 25 fps. Average bit rate: 2Mbps (“Nemo1M” has 1 Mbps). BP: 5% I, 95% P. MP: 4% I, 48% P, 48% B. [online] http://www.sec.upm.es/gdem/en/test_sequences.php.

³ The audio streams have been encoded with MPEG-2 layer II (see note 2).



Fig.5. Test-bench used in real-time tests.

TABLE II.
IP SET-TO-BOX PERFORMANCE DATA.

| CPU% | NEMO | | STAR WARS | | FOOTBALL | | NEMO1M | |
|---------|------|-------|-----------|-------|----------|-------|--------|-------|
| | BP | MP | BP | MP | BP | MP | BP | MP |
| DECODER | 81.2 | 107.5 | 77.0 | 108.7 | 79.2 | 107.9 | 65.8 | 101.6 |
| TOTAL | 98.2 | 124.5 | 94.0 | 125.7 | 96.2 | 124.5 | 82.8 | 118.6 |

IV. CONCLUSIONS

In this paper, the implementation of an H.264 decoder on a low-cost DSP and its integration on a single-chip multi-format STB have been shown. Tests in a real environment have been made with a 600 MHz system clock. These tests show that real-time is achieved for BP and could be achieved for MP with a 720 MHz system clock. Our current work is focused on the assembling of some modules and the implementation of ASO, slice groups and interlaced video.

REFERENCES

- [1] ISO114496-2. Information technology. Coding of audio-visual objects. Part 2: Visual. May 2004.
- [2] ISO114496-10. Information technology. Coding of audio-visual objects. Part 10: Advanced Video Coding. Dec. 2005.
- [3] Ostermann, J. et al. “Video coding with H.264/AVC: tools, performance, and complexity”. IEEE Circuits and Systems Magazine, Vol. 4, Issue 1, pp. 7-28, 2004.
- [4] Philips Semiconductors. Nexperia Media Processors. [online] http://www.nxp.com/products/nexperia/home/products/media_processors/index.html
- [5] Analog Devices. Blackfin processors. [online] <http://www.analog.com/products/blackfin/index.html>
- [6] Texas Instruments. TMS320DM642 DSPs. [online] <http://focus.ti.com/docs/prod/folders/print/tms320dm642.html>
- [7] Y.S. Tung et al. “DSP-Based Multi-Format Video Decoding Engine for Media Adapter Applications”. IEEE Trans. on Consumer Electronics, Vol. 51, Issue 1, pp. 273 - 280, Feb. 2005.
- [8] Vaidyanathan Ramadurai et al. “Implementation of H.264 decoder on Sandblaster DSP”. IEEE International Conference on Multimedia and Expo, 2005. ICME 2005, 5 pp. Jul. 2005.
- [9] Moshe, Y.; Peleg, N. “Implementations of H.264/AVC baseline decoder on different digital signal processors”. 47th International Symposium ELMAR, pp. 37 - 40, Jun. 2005.
- [10] F. Pescador et al. “A Real-Time H.264 BP decoder based on a DM642 DSP”. IEEE International Conference in Signal Processing and Communication. Dubai. Nov 2007. Paper accepted.
- [11] F. Pescador et al. “A DSP based IP set-top box for home entertainment”. IEEE Trans. on Consumer Electronics Vol. 52, Issue 1, Feb. 2006 pp. 254-262.
- [12] DM642 evaluation module. [online] <http://www.spectrumdigital.com>
- [13] Reference Frameworks for eXpressDSP Software: RF5 (SPRA795A-April 2003). [online] <http://focus.ti.com/lit/an/spra795a/spra795a.pdf>.
- [14] AMK 430 AVC encoder. [online] <http://www.ateme.com>